NASA TECHNICAL TRANSLATION

NASA TT F-483

c.1

NASA TT F-483

A METHOD OF DYNAMIC PROGRAMMING AND ITS APPLICATION TO OPTIMIZATION PROBLEMS OF FLIGHT MECHANICS

by W. Schulz and H.-K. Schulze

Paper presented at Wissenschaftliche Gesellschaft für Luft-und Raumfahrt and Deutsche Gesellschaft für Raketentechnik und Raumfahrtforschung, Jahrestagung, Berlin, West Germany, September 14-18, 1964

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1967

A METHOD OF DYNAMIC PROGRAMMING AND ITS APPLICATION TO OPTIMIZATION PROBLEMS OF FLIGHT MECHANICS

By W. Schulz and H.-K. Schulze

Translation of "Die Methode des Dynamic Programming und ihre Anwendung auf Optimierungsaufgaben der Flugmechanik." Paper presented at Wissenschaftliche Gesellschaft für Luft- und Raumfahrt, and Deutsche Gesellschaft für Raketentechnik und Raumfahrtforschung, Jahrestagung, Berlin, West Germany, September 14-18, 1964.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

A METHOD OF DYNAMIC PROGRAMMING AND ITS APPLICATION TO OPTIMIZATION PROBLEMS OF FLIGHT MECHANICS /1[**]

W.Schulz and H.-K.Schulze[*]

Discussion of Bellman's method of programming and its applicability to the numerical solution of optimization problems of flight mechanics. The applicability of the method is illustrated on two practical examples - i.e., optimization of the payload ratio of a multistage rocket and determination of brachistochronic flight paths. Some results of computer calculations are discussed.

Summary /2

The method of dynamic programming, based on the optimization principle established by R.Bellman, is suitable for the treatment of optimization problems in which - for example, due to the existence of secondary conditions in the form of inequalities - the prerequisites for solution with the classical indirect method of the calculus of variations are not met. On hand of examples of optimization of a multistage rocket and determination of brachistochronic flight paths, practical application of the method of dynamic programming is described. Practical experience with the computational procedure and problems of accuracy are discussed in some detail.

Contents

1. Introduction /3

The purpose of this paper, discussing the method of dynamic programming and its application to optimization problems in flight mechanics, is not so much to report novel flight-mechanics data but rather to give a general survey over the applicability of a method which had been developed specifically for the numeri-

[*] German Aeronautics and Space Research Organization, Institute for Flight Mechanics, Braunschweig.

[**] Numbers given in the margin indicate pagination in the original foreign text.

cal solution of optimization problems. This method, which carries the name of "dynamic programming" was developed by the American mathematician Richard Bellman who was involved in this program ever since the beginning of the Fifties and applied it, together with coworkers at the Rand Corp., to numerous problems in widely differing fields of industrial research, national economics, and general engineering. The data obtained in these investigations were published not only in a large number of individual papers but also in two books. The book by Bellman "Dynamic Programming" published in 1957 (Ref.1) contains the theoretical principles; for various types of optimization problems, the formulation is given as a dynamic programming problem with a description of the method of solution. Since the method is intended primarily for handling optimization problems by numerical treatment with the use of a digital computer, the book "Applied Dynamic Programming" published by Bellman in cooperation with Dreyfus (Ref.2) is mainly concerned with computational viewpoints under treatment of practical examples, constantly emphasizing the limits still frequently encountered in practical execution of the computations. An excellent survey over the method is contained in an article by Dreyfus, in the Collective Works "Progress in Operations Research" published by R.L.Ackoff (Ref.3). The method of dynamic programming presupposes that the problem to be solved can be brought to the form of a multistage decision process. Therefore, we will first discuss the necessary conceptual aspects and then formulate the so-called optimality principle by Bellman, which forms the basis for practical application of the method.

The fields in which the method of dynamic programming can be applied include also optimization problems of rocket technology and problems in flight-path optimization. Two simple examples of this type will be used for explaining the method in some detail.

The problem of selecting the masses for the stages of a multistage /4
rocket, which is to reach a prescribed burnout velocity, in such a manner that the liftoff mass of the rocket will remain as small as possible, was specifically treated by Ten Dyke (Ref.4) under application of the method of dynamic programming [see also others (Ref.1, p.145, Problem 55; Ref.2, pp.227-228).

Problems of flight-path optimization such as the determination of the optimal climb technique of a given aircraft and of optimal satellite trajectories were treated in several papers by Bellman and coworkers (Ref.5, 6, 7). In addition, Bellman published a summary report on the determination of optimal trajectories via dynamic programming, in the book by G.Leitmann "Optimization Techniques with Application to Aerospace Systems" (Ref.8). Several papers in this field have also been published in Germany within the past two years. In a Darmstadt thesis, J.Spintzyk (Ref.9) investigated optimal time-to-turn and overtake maneuvers of supersonic aircraft, under application of the method of dynamic programming; H.Friedel and K.Stopfkuchen at the Dornier Co. (Ref.10) applied the method to calculating time-optimal transition flights of VTOL aircraft.

In the last portion of the paper, we will report on computational experience gained on the Siemens 2002 digital computer of the DFL in Braunschweig. To define the basic problems, we selected simple examples that are not burdened with the extensive computational apparatus inherent to flight-mechanics problems but are of interest mainly from the mathematical viewpoint.

2. Basic Concepts; Principle of Optimality

The method of dynamic programming is used for determining the optimal procedure in multistage decision processes. In a more general formulation, this means: A system is to be transferred from an initial state, over a number of steps, into a final state in such a manner that this transfer is optimal in a sense to be defined more accurately in each individual case. The initial state is conceived as represented by a point in a plane (point farthest to the left in Fig.1). Let us now assume a number of possibilities for reaching new states from the initial point. The transition from the initial state to a new state requires a "decision". The totality of the states that can be reached from /5 the initial state via the possible decisions represents the first stage of the process. From each of the states in the first stage, the states of the second stage are reached by a possible decision, and so on. After N steps, the sought final stage should be reached. This results in a plotting of the type shown in Fig.1.

A sequence of decisions, leading from the initial to the final state, is known as a "plan". Of all possible plans, that type is sought which will have a desired optimum property. This is then called the "optimal plan".

At the head of his considerations, Bellman placed the following principle:

Bellman's Optimality Principle

An optimal plan for the decisions has the property that, independent of the type of initial decision and of the type of state present, the remaining decisions again form an optimal plan with respect to the state obtained from the first decision.

This principle can be more or less considered an axiom since it seems reasonably inconceivable that it should not be valid. For example, if one imagines, in a given problem in which the expenditure (expenditure of time or cost) is to be reduced to a minimum, the corresponding expenditure as being determined for each decision, then the expenditure from any intermediate state to the final state must not be greater for the optimal plan than for any other plan which leads from the intermediate state to the final state.

This principle corresponds exactly to the property of the brachistochrone which had furnished the incentive for development of the calculus of variations by the brothers Bernoulli. The brachistochrone has the property of representing the trajectory between an initial point A and an end point E in a vertical plane on which a certain mass point, under the effect of gravity, travels from A to E in a shorter time than by any other path. In addition, the brachistochrone has the property, for each intermediate point, that it represents the /6 curve of steepest descent from there to the end point E.

In Fig.1, only a single initial and a single end state are assumed. There is no objection to permit an arbitrary number of states for the zero stage and for the N-th stage. In the language of the calculus of variations, one then has no longer to do with point-point problems but with initial boundary-point,

point-final boundary, or initial boundary-final boundary problems.

It might happen that, from some state of one stage, the decision possibilities to all states of the next stage are available. However, also the other extreme is conceivable, namely that, starting from each state, one can choose only between two decisions or has only a single possibility of reaching a state of the next stage.

The point of importance for computational application is the fact that the process under consideration consists of a finite number of stages and that, in each case, only a finite number of states is present. Therefore, the method is intended for problems in which the variables assume only discrete values.

However, the method is also directly applicable to continuous problems if these are discretized. Such a procedure is nothing unusual in mathematics. We merely need recall the methods for numerical solution of differential equations in which these are replaced by difference equations. Problems raised in the transition from continuous variables to discrete values, including questions of accuracy, etc., will be discussed later. This transition permits treatment of problems of the calculus of variations, which naturally include flight-path optimizations, using the method of dynamic programming.

3. Optimization of Payload Ratios in a Multistage Rocket

As a typical example for the practical application of the method of dynamic programming, let us analyze an optimization problem from rocket technology. /7
Let us consider an N-stage rocket. Figure 2 contains the notations for the masses of the stages and of the entire rocket on ignition and on burnout of the individual stages. Here, M is the mass of one stage and m is the mass of the total rocket. The superscript gives the number of the stage while the subscript 0 indicates the instant of ignition of a given stage, with the subscript bo denoting the instant of burnout.

The payload of the k-th stage constitutes the remainder of the rocket carried by this particular stage, i.e., $m_0^{(k+1)}$. The actual payload mass can thus be denoted by $m_0^{(N+1)}$. The quotient

$$\eta^{(K)} = m_0^{(K+1)} / m_0^{(K)} \qquad (K = 1, 2, \ldots, N) \tag{3.1}$$

is called the "payload ratio" of the k-th stage. The product of all payload ratios yields the ratio of the actual payload $m_0^{(N+1)}$ to the liftoff mass $m_0^{(1)}$:

$$\eta^{(1)} \eta^{(2)} \cdots \eta^{(N)} = \frac{m_0^{(2)}}{m_0^{(1)}} \frac{m_0^{(3)}}{m_0^{(2)}} \cdots \frac{m_0^{(N+1)}}{m_0^{(N)}} = \frac{m_0^{(N+1)}}{m_0^{(1)}}. \tag{3.2}$$

The "structure factor" of the k-th stage is the ratio of its mass on burnout and on ignition:

$$\varepsilon^{(K)} = M_{bo}^{(K)} / M_0^{(K)} . \qquad (3.3)$$

Since

$$m_0^{(K)} = M_0^{(K)} + m_0^{(K+1)} \qquad (3.4)$$

is valid, we can also write

$$\varepsilon^{(K)} = m_{bo}^{(K)} / (m_0^{(K)} - m_0^{(K+1)}) . \qquad (3.5)$$

Let us next consider the flight of an N-stage rocket in drag-free and gravity-free space. As soon as one stage has burnt out, it separates from the rest of the rocket at the instant of burnout. Simultaneously, the next stage is ignited so that the values for the time and velocity on ignition of a given /8 stage coincide with the burnout values for the time and velocity of the preceding stage:

$$t_0^{(K)} = t_{bo}^{(K-1)} , \qquad V_0^{(K)} = V_{bo}^{(K-1)} \qquad (K = 2, \ldots, N). \qquad (3.6)$$

Let the exhaust velocity of the combustion gases, for the k-th stage, have the value $c^{(k)}$.

The question here is: If the structure factors $\varepsilon^{(1)}, \varepsilon^{(2)}, \ldots, \varepsilon^{(N)}$ and the exhaust velocities $c^{(1)}, c^{(2)}, \ldots, c^{(N)}$ as well as the ratio of payload to liftoff mass

$$m_0^{(N+1)} / m_0^{(1)} = a^* \qquad (3.7)$$

are given, how does the payload ratio $\eta^{(1)}, \eta^{(2)}, \ldots, \eta^{(N)}$ have to be selected to have the burnout rate of the last stage become a maximum?

Using the fundamental equation of rocket technique, the velocity increment from ignition to burnout can be calculated for each stage; by summation, the velocity at burnout of the N-th stage will then be obtained as

$$V_{bo}^{(N)} = L^{(1)}(\eta^{(1)}) + L^{(2)}(\eta^{(2)}) + \cdots + L^{(N)}(\eta^{(N)}) \qquad (3.8)$$

with

$$L^{(K)}(\eta^{(K)}) = c^{(K)} \left| \ln\left[\varepsilon^{(K)} + (1-\varepsilon^{(K)})\eta^{(K)}\right] \right| \qquad (K = 1, 2, \ldots, N). \qquad (3.9)$$

Accordingly, one has to define at what selection of $\eta^{(1)}, \eta^{(2)}, \ldots, \eta^{(N)}$ the function

$$f(\eta^{(1)}, \eta^{(2)}, \ldots, \eta^{(N)}) = \sum_{K=1}^{N} L^{(K)}(\eta^{(K)}) \qquad (3.10)$$

becomes a maximum, in which case the auxiliary condition exists that - according to eqs.(3.2) and (3.7) - the product of the payload factors has the prescribed value a*:

$$\eta^{(1)}\eta^{(2)}\ldots\eta^{(N)} = a^{*}. \tag{3.11}$$

The sought maximum is a function of N and a*: /9

$$F_N(a^*) = \underset{(\eta^{(K)})}{Max}\left[\sum_{K=1}^{N} L^{(K)}(\eta^{(K)})\right], \tag{3.12}$$

in which case the variables $\eta^{(k)}$ are located in the interval

$$0 \leqq \eta^{(K)} \leqq 1 \qquad (K = 1, 2, \ldots, N) \tag{3.13}$$

and are subject to the auxiliary condition (3.11).

Instead of the special given value a*, let us now consider, in a general manner, the function $F_N(a)$ for an arbitrary a which, just as a*, is subject to the condition

$$0 \leqq a \leqq 1. \tag{3.14}$$

For $F_N(a)$ a recursive formula is to be established. For N = 1, we have

$$F_1(a) = \underset{\eta^{(1)}=a}{Max}\; L^{(1)}(\eta^{(1)}) = L^{(1)}(a). \tag{3.15}$$

Then, seeking for the maximum (3.12) is conceived as an N-stage decision process. Let us imagine the first N-1 decisions as already being taken, so that the last step must be made. For the first N-1 variables $\eta^{(k)}$ the following is valid:

$$\eta^{(1)}\eta^{(2)}\ldots\eta^{(N-1)} = \frac{a}{\eta^{(N)}} . \tag{3.16}$$

From the initial state to the state after N-1 decisions, an optimal path had to be used in which case the relation (3.16) as auxiliary condition had to be /10 satisfied. The solution for this reads $F_{N-1}(a/\eta^{(N)})$. To this must be added, as the final step, the summand $L^{(N)}(\eta^{(N)})$, from which it follows that

$$F_N(a) = \underset{\eta^{(N)}}{Max}\left\{L^{(N)}(\eta^{(N)}) + F_{N-1}\left(\frac{a}{\eta^{(N)}}\right)\right\}. \tag{3.17}$$

In general, the following is valid for each k = 2, ..., N:

$$F_K(a) = \underset{a \leq \eta \leq 1}{Max} \left\{ L^{(K)}(\eta) + F_{K-1}\left(\frac{a}{\eta}\right) \right\}. \qquad (3.18)$$

Under consideration of eq.(3.15), the formula (3.18) offers a possibility of recursively calculating the values $F_k(a)$ for $k = 1, 2, \ldots, N$. By means of the functional equation (3.18), the problem has been made accessible to handling on a digital computer. The fact that, in this case, a number of problems may occur in the practical solution will be further discussed in Section 5.

According to eq.(3.18), the maximum of the function

$$\varphi(\eta) = L^{(K)}(\eta) + F_{K-1}\left(\frac{a}{\eta}\right) \qquad (3.19)$$

must be sought in the interval $a \leq \eta \leq 1$. Since, in the case under consideration, $L^{(K)}(\eta)$ because of eq.(3.9) is known in analytic form, the maximum can be obtained from the condition $\varphi'(\eta) = 0$ provided that the solution obtained from this condition is located in the prescribed interval. However, this need not necessarily be the case. Then, the function (3.19) assumes its maximum value along the boundary of the intervals which means that, at given values for the exhaust velocities and the structure factors, maximum burnout velocity is reached with a rocket of less than N stages.

In contrast to the problem discussed here, the maximum can generally not /11 be determined by analytical means and, possibly, may require extensive numerical calculations on a digital computer. We will return to this particular problem in Section 5.

4. Optimum Climb of an Aircraft

A given aircraft, flying at a speed V_0 at an altitude h_0, is to be brought, in a climb, to an altitude $h_e > h_0$ and to a speed of $V_e > V_0$ in the shortest possible time. The thrust S and the aerodynamic drag W must depend only on the altitude h and the speed V and must be known functions of these variables. The variation in aerodynamic drag with the angle of attack and the influence of weight reduction due to fuel consumption will be disregarded here. For the path inclination angle γ, the values $0 \leq \gamma \leq \gamma_{max} \leq 90^\circ$ will be admitted. The goal is to establish the optimal climb program and to calculate the required time of flight.

Here, the following equations of motion are valid:

$$m\dot{V} = S - W - mg\sin\gamma, \qquad (4.1)$$

$$\dot{h} = V\sin\gamma \qquad (4.2)$$

with the boundary conditions at the time $t = 0$

$$V(0) = V_0 \quad , \quad h(0) = h_0 \tag{4.3}$$

and, at the unknown time t_e,

$$V(t_e) = V_e \quad , \quad h(t_e) = h_e \, , \tag{4.4}$$

and, in addition, it is assumed that

$$t_e = Min. \tag{4.5}$$

To obtain the functional equation, permitting an application of the method of dynamic programming, we will decompose the velocity interval from V_0 to V_e into partial intervals of a size ΔV and the altitude range from h_0 to h_e into subsegments of a size Δh. From the differential equations (4.1) and (4.2), the relations between ΔV, Δh and the time interval Δt are obtained: /12

$$\Delta V = \left(\frac{S - W}{m} - g \sin\gamma \right) \Delta t \, , \tag{4.6}$$

$$\Delta h = V \sin\gamma \, \Delta t \, , \tag{4.7}$$

$$\Delta V = \left(\frac{S - W}{m V \sin\gamma} - \frac{g}{V} \right) \Delta h \, . \tag{4.8}$$

As independent variable, the altitude increment Δh can be selected for a path segment with $\gamma > 0$ while, for a horizontal segment along which the speed of the aircraft is increased, the velocity increment ΔV can be used. Path segments at constant altitude and constant speed are of no interest for our consideration. In addition, no basic difficulties are produced by admitting also path segments with dives at which the angle of inclination becomes negative and the speed is increased under loss of altitude.

Let the minimum time required by the aircraft to pass from the state h, V to the final state h_e, V_e be f(H, V).

The time required for climbing from the state h, V under the angle γ by an amount Δh, is denoted by t(h, V, γ, Δh) respectively by t(h, V, γ, ΔV), depending on whether Δh or ΔV constitutes the independent variable.

On the basis of Bellman's optimality principle, the equation

$$f(h,V) = \underset{0 \leq \gamma \leq \gamma_{max}}{Min} \left[t(h,V,\gamma,\Delta h) + f(h+\Delta h, V+\Delta V) \right] \tag{4.9}$$

respectively

$$f(h,V) = \underset{0 \leq \gamma \leq \gamma_{max}}{Min} \left[t(h,V,\gamma,\Delta V) + f(h+\Delta h, V+\Delta V) \right] \tag{4.10}$$

is obtained. Here, the second summand on the right-hand sides represents the /13 minimum time required for passing from the state h + Δh, V + ΔV to the final state h_e, V_e. In eq.(4.9), ΔV - in accordance with eq.(4.8) - must be expressed by Δh while, in eq.(4.10), Δh must be expressed by ΔV.

For the first summand on the right-hand side of eq.(4.9), we have

$$t\,(h,V,\gamma,\Delta h) = \frac{\Delta h}{V\sin\gamma}\ , \tag{4.11}$$

and, in the case of eq.(4.10), accordingly

$$t\,(h,V,\gamma,\Delta V) = \frac{\Delta V}{\frac{S-W}{m} - g\sin\gamma}\ . \tag{4.12}$$

Equations (4.9) and (4.10) represent the recursive formulas for application of the method of dynamic programming. After assuming a fixed value for Δh resp. ΔV one starts with the final state h_e, V_e, for which

$$f(h_e\ , V_e\ ) = 0. \tag{4.13}$$

For the values h_e, V_e the thrust and drag are calculated, which are then taken as constant for the next step. After this, a sequence of values γ_1, γ_2, ..., γ_i are selected for the path inclination angle γ, calculating the right-hand sides of eq.(4.9) resp. (4.10) for this sequence and thus obtaining the times t_1, t_2, ..., t_i for the i states of the (n - 1)-th stage. From eqs.(4.6) resp. (4.7), the corresponding velocity resp. altitude is obtained in each case. Determination of the minimum of the calculated values will then furnish the value f(h, V). In this manner, it is possible to pass from each state of a given stage to the states of the preceding stage.

5. Remarks on Computational Handling of the Method

The method of dynamic programming, because of the enormous computational effort, can generally not be applied without the use of electronic computers. Consequently, the above statements will be supplemented from the computational viewpoint, in which case we will restrict our discussion of the examples to the previously treated problem of the payload ratio of a multistage rocket as well as to a few problems of the calculus of variations. On hand of these examples, the computational handling of the dynamic programming and the difficulties and problems occurring in this particular computational method will be demonstrated. /14

a) First Example

The method of dynamic programming can be explained on the following example:

The function

$$R(y^{(0)}, y^{(1)}, \ldots, y^{(N)}) = \sum_{K=1}^{N} f(y^{(K)}, y^{(K-1)}) \qquad (5.1)$$

dependent on the N + 1 variables $y^{(o)}$, $y^{(1)}$, ..., $y^{(N)}$ with the boundary conditions

$$y^{(N)} = y_P \quad , \qquad y^{(0)} = y_Q \qquad (5.2)$$

is to be minimized by suitable choice of the variables $y^{(k)}$ (k = 1, ..., N - 1). Here, we make the restriction that the $y^{(k)}$ variables can assume only certain discrete values $y_i^{(k)}$.

The calculation can proceed stepwise as an N-stage process. We have

$$F_1(y_i) = f(y_i^{(1)}, y_Q), \qquad (5.3)$$

$$F_K(y_i) = \underset{(y_j^{(K-1)})}{Min} \left\{ f(y_i^{(K)}, y_j^{(K-1)}) + F_{K-1}(y_j) \right\} \qquad (K = 2, \ldots, N). \qquad (5.4)$$

The graphic representation of an optimization plan (Fig.3) indicates the calculation procedure. The calculation, according to eqs.(5.3) and (5.4), consists of two parts.

a) Calculation of the initial column: For each $y_i^{(1)}$ of the first stage, the corresponding value $F_1(y_i)$ is calculated according to eq.(5.3) and then stored.

b) Extremization sequence (considering only the k-th stage): For each $y_i^{(k)}$ of the k-th stage, in accordance with eq.(5.4), that $y_j^{(k-1)}$ of the (k - 1)-th stage is sought which minimizes the contents of the brace in eq.(5.4). The obtained value $y_j^{(k-1)}$ and the function value $F_k(y_i)$ are stored in the memory.

If the extremization sequence is continued to the N-th stage, exactly /15 one line segment will be obtained for each state $y_i^{(N)}$ of the N-th stage, relative to the prescribed final state $y^{(o)} = y_Q$.

Thus, extremization of the N-th stage has yielded a result going beyond the required problem formulation, i.e., without excessive additional computational effort a special boundary-point problem has been solved instead of a point-point problem. In numerous practical cases, such a supplementary effort is well worth it.

b) Memory Requirement and Computation Time

In calculating the $y_{(j)}^{(k-1)}$ of the (k - 1)-th stage and the $F_k(y_i)$ of the k-th stage, in accordance with eq.(5.4) only the $F_{k-1}(y_j)$ of the (k - 1)-th stage but not the function values of the preceding stages were required.

On denoting the maximum number of possible states per stage by h, it is sufficient to reserve 2h memory registers for the functional values.

If exactly h states are permitted in each stage, the following will apply for the simple example treated here:

Number of memory locations for data = $(N+1)h$;

Calculating time, approximately proportional to $(N-1)h^2 + h$.

If such a problem depends not only on one but on n variables which all can assume h states in each stage, then the following rule of thumb applies:

Number of memory locations for data = $(N+1)h^n$,

Calculating time, approximately proportional to $(N-1)h^{2n} + h^n$.

The difficulties arising from these rules of thumb will be discussed later in the text.

c) Second Example

The programming technology of a given problem, by means of dynamic programming, will be explained on the example of calculating the payload ratios for obtaining as high as possible a burnout velocity in an N-stage rocket. /16

According to the statements in Section 3, this optimization problem

$$F_N(a^*) = \underset{\eta^{(1)}\cdots\eta^{(N)}=a^*}{Max} \left\{ \sum_{K=1}^{N} L^{(K)}(\eta^{(K)}) \right\} \qquad (5.5)$$

can be written in the form of an N-stage process

$$F_1(a) = L^{(1)}(a)\,, \qquad (5.6)$$

$$F_K(a) = \underset{a^* \leq a \leq \eta \leq 1}{Max} \left\{ L^{(K)}(\eta) + F_{K-1}\left(\frac{a}{\eta}\right) \right\} \qquad (K = 2, 3, \ldots, N) \qquad (5.7)$$

and thus solved by means of dynamic programming.

In each stage of the optimization process, only a limited number of states can be permitted, i.e., a, η, and a/η can assume only a limited number of discrete values.

Introducing the variation interval δ and selecting δ in such a manner that the expressions $1/\delta = h$ and (if at all possible) also a^*/δ are integers, we will assume that a and η, from here on, can take only the values $i\delta$ with $i = 1, 2, \ldots, h$. In the problem under discussion, it is preferable to select δ in the order of magnitude 0.002 - 0.01.

If $a = i\delta$ and $y = j\delta$, the above quotient $a/\eta = i/j$ will generally be a multiple of δ. For this reason, it is preferable to select the value closest to a/η from the permissible data reserve, using the formula

$$\frac{a}{\eta} \approx \delta \cdot \text{ integer } (i/j\delta + 1/2),$$

where integer x denotes the maximum whole number smaller than or equal to x. /17 This completely discretizes the problem. In addition, the three variables (stage, state, decision) occur as subscripts (k, i, j) and thus permit a convenient computational data processing.

If one uses the following expression, in analogy to the conventional indexing of matrices with the use of the ALGOL notation:

$$F_K(i\delta) = F[i, K]$$

and with the use of the Zuse "yields" sign, then the machine-sensible indexed form of the problem will be obtained:

$$L^{(1)}(i\delta) \Rightarrow F[i, 1], \tag{5.8}$$

$$\underset{1 \leq j \leq h}{Max}\left\{ L^{(K)}(j\delta) + F\left[\text{integer } (i/j\delta + 1/2), K-1\right]\right\} \Rightarrow F[i, K]$$

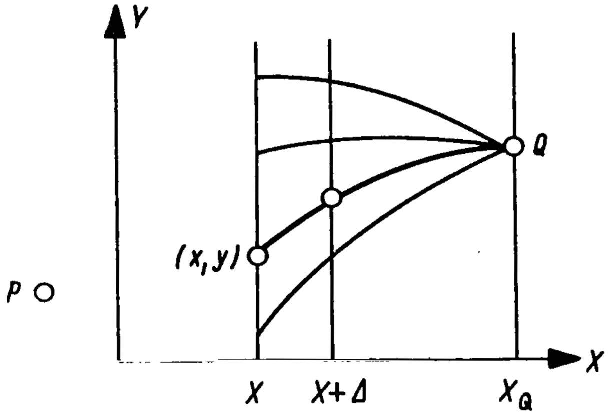
$$(K = 2, \ldots, N;\ i = 1, \ldots, h). \tag{5.9}$$

The course of the extremization calculation (initial column and extremization sequence) is shown in the flow chart (Fig.4). [In the flow chart, the contents of the trapezoids above a given box have the meaning of an execution instruction. For example, k = 2(1)N means: Execute the contents of the box continuously for all k of k = 2, with an interval of 1 to k = N.]

The calculations for this problem were performed on the Siemens 2002 digital computer of the DFL with an ALGOL (algorithmic language) program. The computation time until establishment of optimal programming for $\delta = 1/100$ and N = 3 was about 9 min.

In view of the discretization, it is obvious that a certain error must be tolerated. For $a^* = 1/100$, this error - constituting the deviation of the product of the calculated η from the prescribed value a^* - was between 10 and 15% while, for $a^* = 1/10$, it was between 2 and 5%. The possibilities for reducing this excessive and impermissible error will be discussed later in the text.

d) Third Example /18



The difficulties produced by discretization are even more pronounced in problems of the calculus of variations. In the simplest case, the problem formulation in the calculus of variations reads as follows:

The integral

$$T(y) = \int_{P(x_P;\, y_P)}^{Q(x_Q;\, y_Q)} f(x, y, y')\,dx \qquad (5.10)$$

is to be minimized by suitable selection of a continuous function y(x).

Disregarding at first the left-hand boundary value $P(x_P;\, y_P)$, one selects an initial value y at an arbitrary point x of the interval $<x_P,\, x_Q>$ (again arbitrary) and then seeks the extremal via the right-hand boundary value Q. Since x and y are arbitrary, a field of extremals with a field function described by

$$F(x, y) = \int_{(x;\, y)}^{Q} f(x, y, y')\,dx \qquad (5.11)$$

is obtained. To obtain an argument permitting an analytic description and a computational treatment, it is necessary to consider a second point x + Δ. Based on the Bernoulli-Euler concept that a randomly selected segment of the extremals must also have extremal properties, the field function F(x, y) can be expressed as follows: /19

$$F(x, y) = \underset{x \leq y \leq x+\Delta}{Min} \left\{ \int_{x}^{x+\Delta} f(x, y, y')\,dx + F(x+\Delta,\, y(x+\Delta)) \right\}. \qquad (5.12)$$

This Bellman argument can be utilized in two different manners:

If the boundary transition Δ → 0 is performed analytically, the following well-known relations of the calculus of variations are obtained after a few transformations:

> the Euler differential equation; the Weierstrass and Legendre conditions;
> the Erdmann corner condition; and the transversatility condition.

For the computational treatment, we must be content with the selection of a finite Δ. If, at given Δ, the number of stages N is chosen such that the abscissa distance from the zero stage to the first stage is not smaller than Δ but smaller than 2Δ and if, again, the variation interval δ is introduced, and using the argument

Stage: $$x^{(K)} = x_p + (N-K)\Delta\,, \qquad (5.13)$$

State: $$y_i^{(K)} = y_p + i\delta\,, \qquad (5.14)$$

Decision: $$y_j^{(K-1)} = y_p + j\delta \qquad (5.15)$$

then we obtain the discretized and indexed instructions

$$\int_{(x^{(1)}; y_p+i\delta)}^{(x_Q; y_Q)} f(x,y,y')dx \Rightarrow F[i,1]\,, \qquad (5.16)$$

/20

$$\underset{(j)}{Min}\left\{\int_{(x^{(K)}; y_p+i\delta)}^{(x^{(K-1)}; y_p+j\delta)} f(x,y,y')dx + F[j,K-1]\right\} \rightarrow F[i,K]. \qquad (5.17)$$

The course of the computation corresponds to that of the above problem.

We computed a number of point-point respectively boundary-point problems with ALGOL programs on the Siemens 2002 computer.

Figure 5 shows the extremal field y = y(x, Q) of the "brachistochrone problem"

$$T = \underset{(y)}{Min}\int_{(3;y)}^{(24;12)} \frac{\sqrt{1+y'^2}}{v(y)}\,dx \qquad \text{with} \quad v(y) = y \qquad (5.18)$$

in the search zone 8.4 ≤ y ≤ 15.9, with the abscissa interval of Δ = 1 and the variation interval of δ = 0.3. The integrations were carried out with the trapezoidal method. The computation time was 14 min. This extremal field distinctly proves that, from each point on the left-hand boundary, a unique path leads to the end point Q.

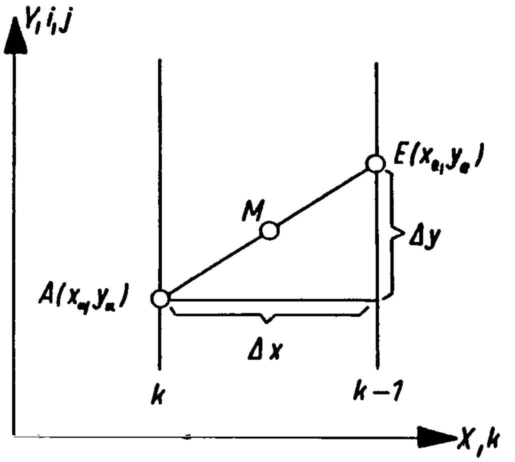


Figure 6 shows three extremals of the same problem but calculated with differing Δ and δ. The exact extremals are given as solid lines, while the calculated points are indicated by circles which, for the extremals Nos.2 and 3, are interconnected by broken lines. The extremal No.2 is identical with the extremals shown as heavy lines in Fig.5.

The difference in absolute amount (exact y-value minus calculated y-value) in these examples is always smaller than one variation interval δ. The ratio of the calculating times agrees well with the above-indicated rule of thumb. It is obvious that the improvement in accuracy results in a considerable increase in calculating time.

e) Integration Method

In the variational problems considered here, the following integration methods were investigated for calculating the integrals over the segments of the fields:

Trapezoidal Method: /21

$$\int_A^E f(x,y,y')\,dx \approx f(x_m,y_m,v)\,\Delta x. \qquad (5.19)$$

Here, v = Δy/Δx.

Simpson Method:

$$\int_A^E f(x,y,y')\,dx \approx [f(x_a,y_a,v)+4f(x_m,y_m,v)+f(x_e,y_e,v)]\,\frac{\Delta x}{6}\cdot \qquad (5.20)$$

Linear Exact Integration:

$$\int_A^E f(x,y,y')dx = \int_A^E f(x, y_a + (x-x_a)v, v)dx = F(x_e, y_e) - F(x_a, y_a). \tag{5.21}$$

This method, however, can be applied only to functions that can be exactly integrated by means of the linear argument

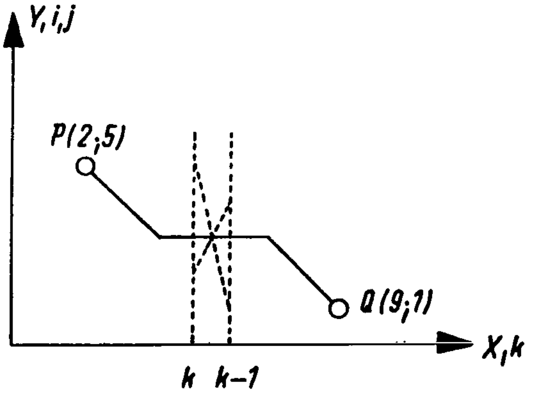
$$y = y_a + (x - x_a)v \tag{5.22}$$

Of these three methods, the trapezoidal method generally requires the least computational effort but is frequently too inaccurate.

In the variational problem

$$T = \underset{(y)}{Min} \int_{(2;5)}^{(9;1)} (y'+1)^2 (y-3)^2 dx, \tag{5.23}$$

whose exact solution is shown in the accompanying sketch, the trapezoidal /22 method fails completely. This is clearly indicated when studying the k-th stage. Not only the exact solution but also all straight lines that intersect the exact solution at the interval center cause the integral to become zero in this stage.



Conversely, the Simpson method yielded satisfactory results in this example even when Δ and δ were so selected that - on the one hand - y never exactly assumed the value 3 and - on the other hand - Δy/Δx never exactly assumed the value -1.

In the variational problem

$$T = \underset{(y)}{Min} \int_{(-1;0)}^{(+1;0)} \frac{\sqrt{1+y'^2}}{y} dx, \tag{5.24}$$

which has no solution at the prescribed boundary points, the trapezoidal method did yield a solution, but an erroneous one.

In the classical brachistochrone problem

$$T = \underset{(y)}{Min} \int_{(0;0)}^{(2\pi;0)} \sqrt{\frac{1+y'^2}{y}}\, dx, \qquad (5.25)$$

which, in 1696, furnished the incentive for development of the calculus of variations, the Simpson method fails as soon as P and Q come to lie on the abscissa. The trapezoidal method produces inaccurate solution while the method of linear exact integration yields a satisfactory solution. The inaccurate solution, /23 when using the trapezoidal method, is dependent mainly on the integration results in the first and last stage. In these two stages, the integral results of the trapezoidal method differ from those obtained by linear exact integration, by a factor of $\sqrt{2}$.

The Gaussian integration method was not investigated at first since, because of the calculation with decimals instead of with proper fractions, the great advantages of this method are canceled.

The question as to the most suitable integration method for a given case cannot be answered in a general manner. Decisions must be made from case to case.

f) Miscellaneous Notes

In addition to selecting an integration method adapted to the problem or, more generally, making a logical selection of the computation method for the functional, it is of importance to calculate all functional components depending only on the stage k in the k-cycle and all functional components depending only on the state i, in the i-cycle. In our examples, this led to a time factor up to 1/50.

Another important point is the choice of the ratio δ/Δ. Fully calculated examples show that, at fixed δ, the result T and the path y(x) will worsen at decreasing Δ. The ratio should not be greater than 1/10 so as to permit a fine gradation of y'. A reduction in the ratio to less than 1/100, conversely, leads to little gain and generally is of no advantage because of the excessive computational effort.

A decisive point in dynamic programming is a logical delimitation of the search zone. The selection, respectively the reduction, of the search zone must take place only after careful physical and mathematical considerations. This also offers the possibility of considerably improving the results of the rocket-staging problem. However, if the limits are drawn too narrow, erroneous results may occur. Figure 7 shows three types of errors, in cases in which the lower limit of the search zone was selected too narrow:

(lim) = the calculated extremal runs along the lower limit;
(osc) = the calculated extremal oscillates along the lower limit;
(rel) = the calculated extremal assumes a relative minimum in the zone.

These three types of extremal falsification were demonstrated by excessively narrowing the search zones for the problem shown in Fig.8. /24

Especially in the case of (rel) is it possible that the calculation leads to a completely erroneous result, despite considerable refining of δ.

Finally, we should mention a physically meaningless but computationally interesting variational problem, which was solved by the method of dynamic programming, with satisfactory accuracy (see Fig.8).

The solution of the problem

$$T = \underset{(y)}{Min} \int_{P}^{Q} (y'+1)^2 (y - \sin x)^2 dx \qquad (5.26)$$

can be given in a closed form whenever the extremal can be piecewise continuously composed of straight-line segments with a slope of -1 in the x-direction and of segments of the sine curve. This holds for the extremal No.1.

The above-mentioned diminution of the search zone was applied to three examples. Too narrow a selection of the limits for the search zone led to the mentioned extremal falsifications.

Finally, it should be noted that the optimization method "dynamic programming" has been successfully used in numerous problems since it is more comprehensive than the other known optimization methods. On the other hand, its limits are clearly defined; they are produced primarily by the large memory requirement and the excessive calculating time.

REFERENCES

1. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, N.J., 1957.
2. Bellman, R.E. and Dreyfus, S.E.: Applied Dynamic Programming. Chapt.VI: Optimal Trajectories, pp.207-233, Princeton University Press, Princeton, N.J., 1962.
3. Dreyfus, S.: Dynamic Programming. In: R.L.Ackoff, Progress in Operations Research. Vol.1, pp.211-242, John Wiley & Sons, New York/London, 1961. /25
4. Ten Dyke, R.P.: Computation of Rocket Step Weights to Minimize Initial Gross Weight. Jet Propulsion, Vol.28, pp.338-340, 1958.
5. Cartaino, T.F. and Dreyfus, S.E.: Application of Dynamic Programming to the Airplane Minimum Time-to-Climb Problem. Aeron. Eng. Rev., Vol.16, No.6, pp.74-77, 1957.
6. Bellman, R. and Dreyfus, S.: An Application of Dynamic Programming to the Determination of Optimal Satellite Trajectories. J. Brit. Interplanet. Soc., Vol.17, pp.78-83, 1959.
7. Smith, F.T.: Optimization of Multistage Orbit Transfer Processes by Dynamic

Programming. ARS (Am. Rocket Soc.) J., Vol.31, pp.1553-1559, 1961.

8. Bellman, R.: On the Determination of Optimal Trajectories via Dynamic Programming. In: G.Leitmann, Optimization Techniques with Applications to Aerospace Systems. pp.281-290, Academic Press, New York/London, 1962.

9. Spintzyk, J.: Investigations on Time-Optimal Turn and Overtake Maneuvers in Supersonic Aircraft (Untersuchungen über zeitoptimale Wende- und Einholmanöver von Überschallflugzeugen). Yearbook 1962 of the Wiss. Ges. Luft- u. Raumf., pp.202-217.

10. Friedel, H. and Stopfkuchen, K.: A Method for Calculating Time-Optimal Transition Flights of VTOL Aircraft (Ein Verfahren zur Berechnung zeitoptimaler Übergangsflüge von VTOL-Flugzeugen). Yearbook 1963 of the Wiss. Ges. Luft- u. Raumf., pp.85-96.

/26

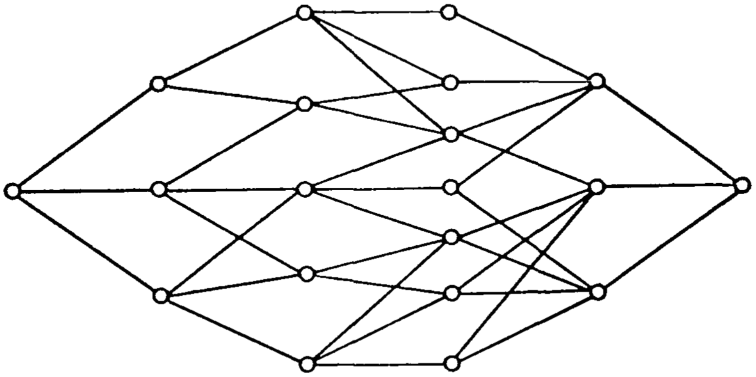


Fig.1 Multistage Decision Process.

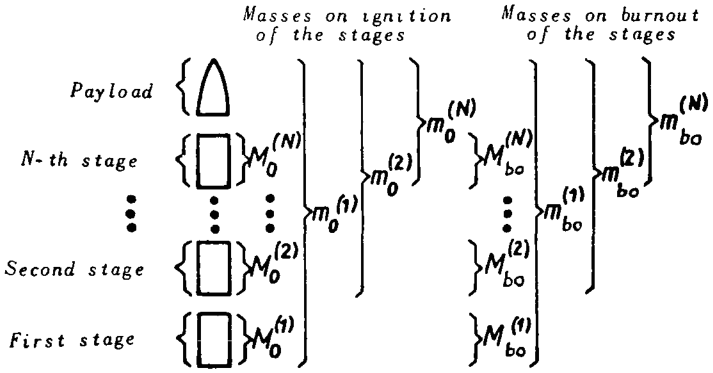


Fig.2 Multistage Rocket.

/27

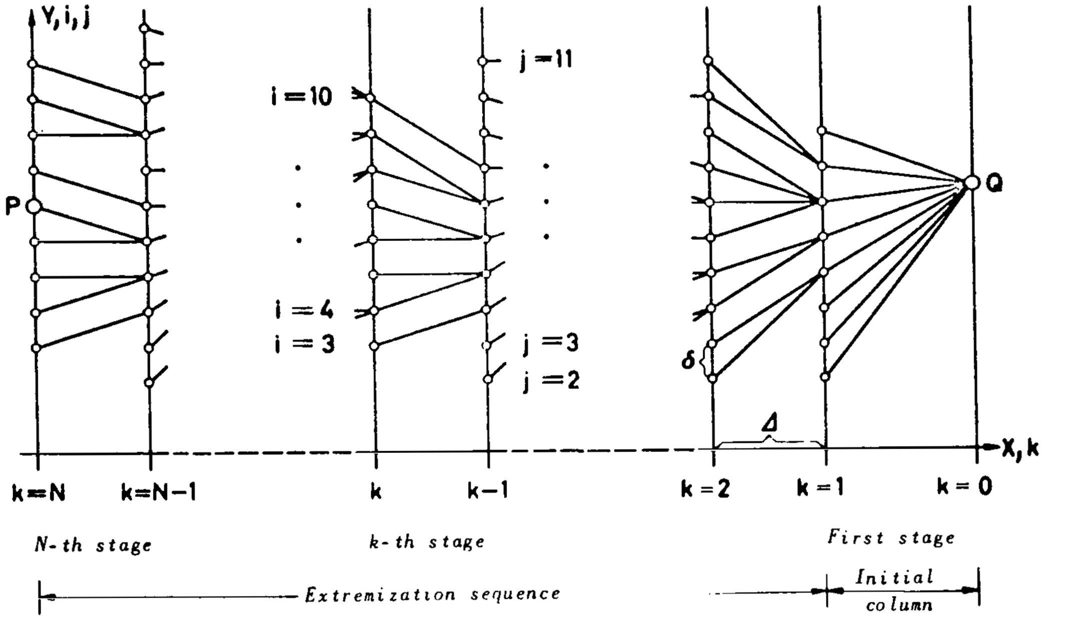


Fig.3 Graphic Representation of an Optimization Plan.

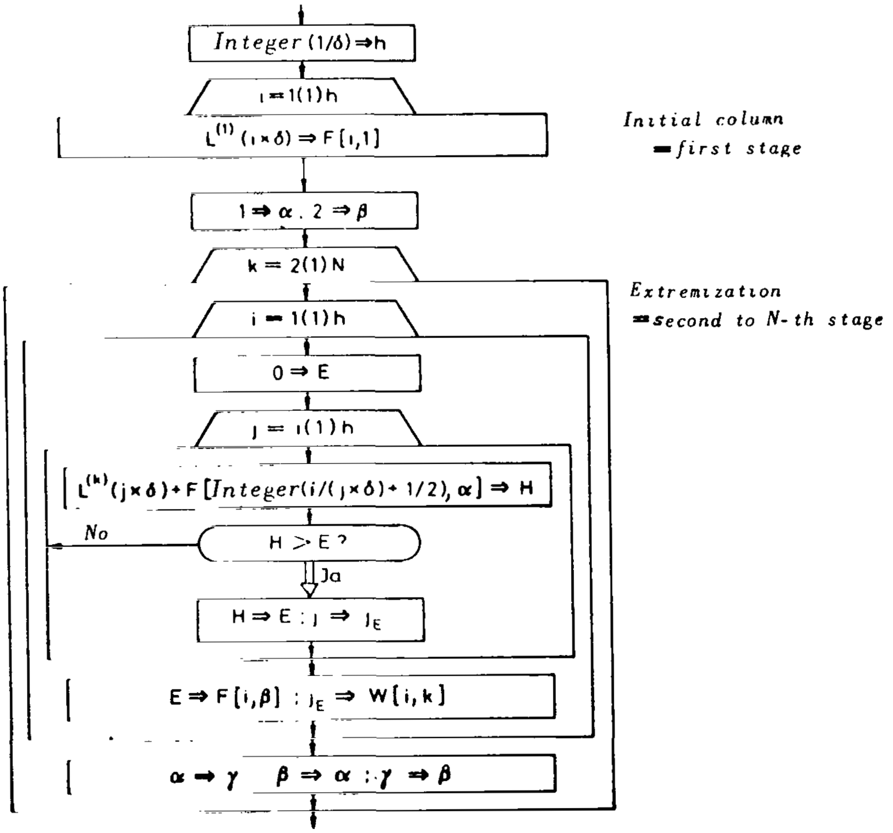


Fig.4 Flow Chart: Extremization Sequence.

/28

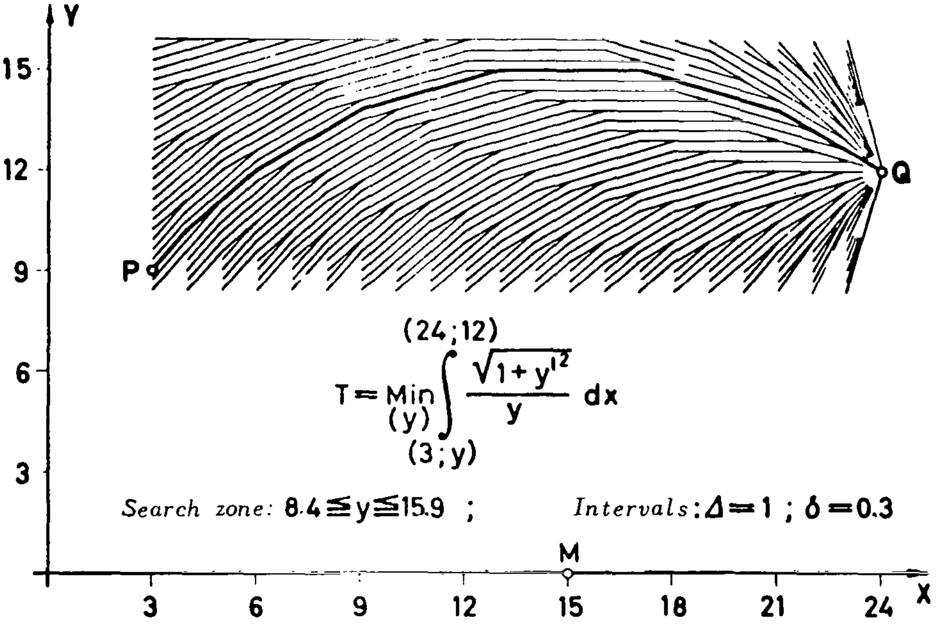
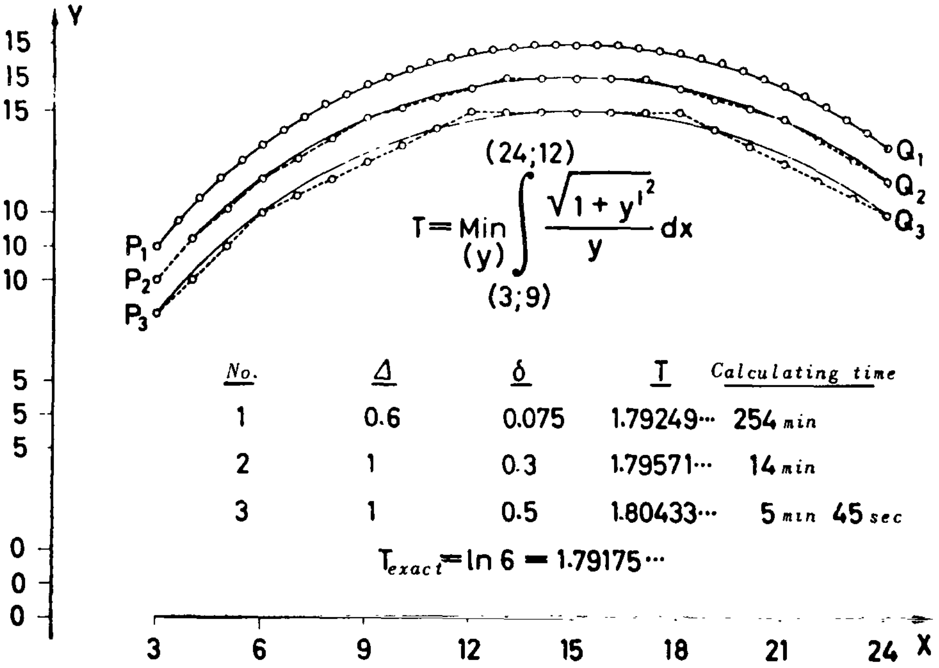


Fig.5 Extremal Field.



No.	Δ	δ	T	*Calculating time*
1	0.6	0.075	1.79249···	254 *min*
2	1	0.3	1.79571···	14 *min*
3	1	0.5	1.80433···	5 *min* 45 *sec*

$T_{exact} = \ln 6 = 1.79175\cdots$

Fig.6 Extremal Calculation with Various Intervals.

/29

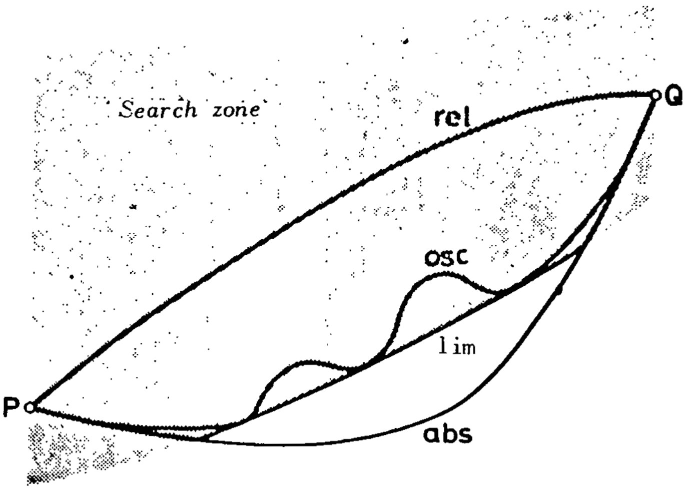
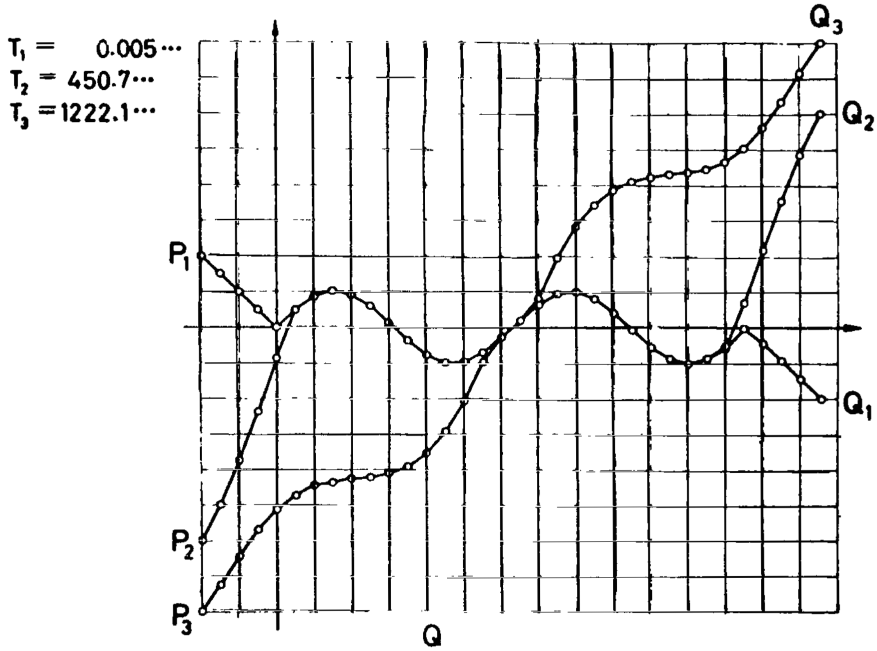


Fig.7 Extremal Falsifications.



$$T = \underset{(y)}{\mathrm{Min}} \int_P^Q (y'+1)^2 (y-\sin x)^2 \, dx$$

Fig.8 Three Extremals.

Translated for the National Aeronautics and Space Administration by the
O.W.Leibiger Research Laboratories, Inc.

NASA-Langley, 1967 —— 1 F-483

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546